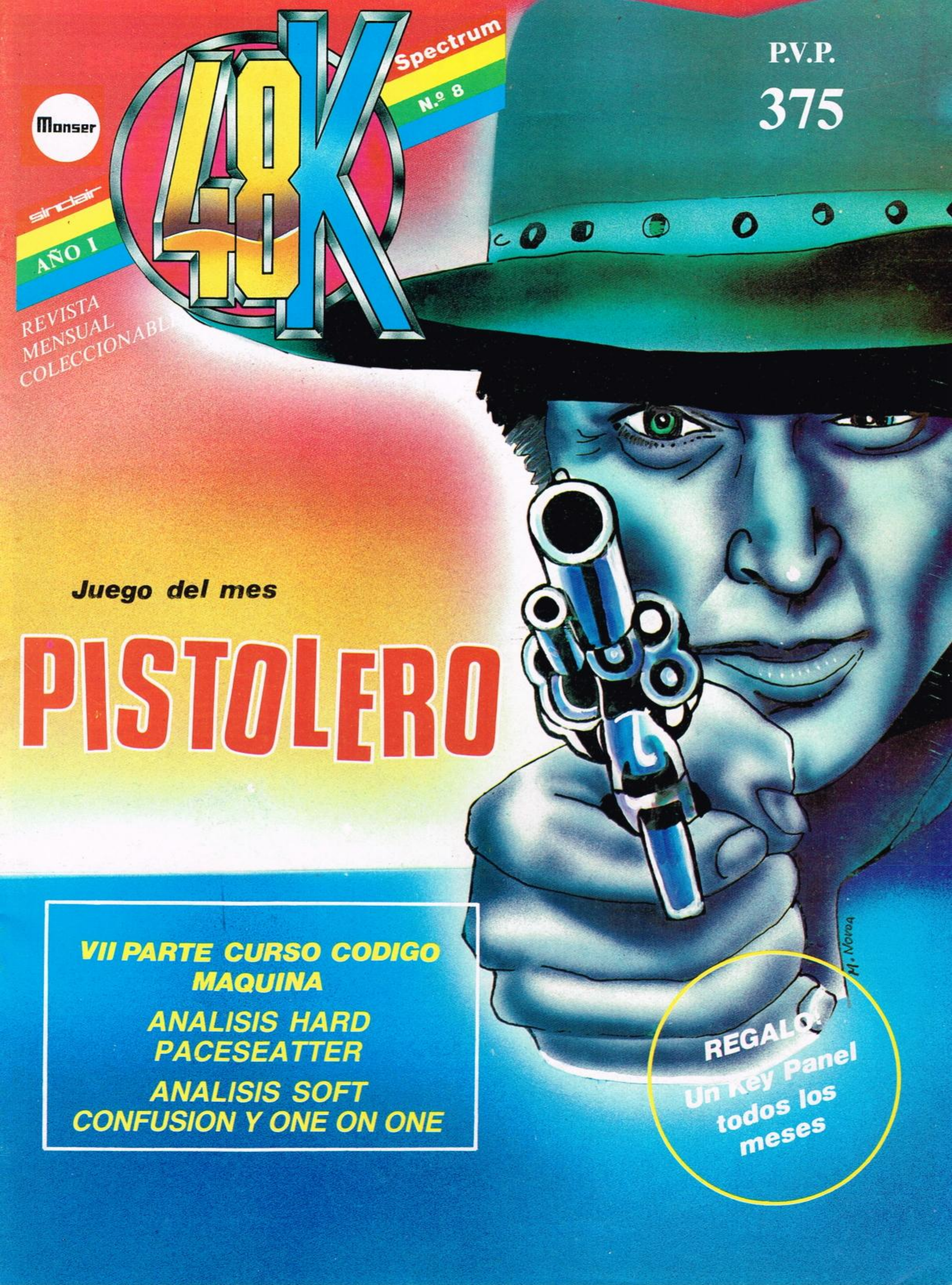

Monser
48K
Spectrum
N.º 8
sinclair
AÑO I
REVISTA
MENSUAL
COLECCIONABLE
P.V.P.
375
Juego del mes
PISTOLERO
VII PARTE CURSO CODIGO MAQUINA
ANALISIS HARD PACESEATTER
ANALISIS SOFT CONFUSION Y ONE ON ONE
REGALO:
Un Key Panel todos los meses
M. Novoa


C-30 CASSETTE ESPECIAL PARA ORDENADOR
La más cargante
530 CASSETTE
ESPECIAL PARA
ORDENADOR
MONSER
SHIFT
C-30
530 CASSETTE ESPECIAL PARA ORDENADOR
C-30



Sumario:

AÑO I - N.° 8 - 1985

DIRECTOR
José Nieto Rubio

COORDINADOR
Félix Santamaría Avila

SUPERVISOR SOFTWARE
Gustavo Cano Muñoz

DISEÑO
Angélica Arce

REDACCION Y COLABORADORES
Victoria Aguilar
Agustín Barcos
Javier González
Mario Alvarez

PORTADA
Mauro Novoa

EDITA
MONSER, S. A.

DIRECTOR EDITORIAL
J. L. Cano Regidor

REDACCION, ADMINISTRACION Y PUBLICIDAD
Argos, 9
28037-MADRID
Tel. 742 72 12/96

PUBLICIDAD Y SUSCRIPTORES
Yolanda Bardillo

FOTOCOMPOSICION
CRISOL, S. A.
Virgen del Val, 48

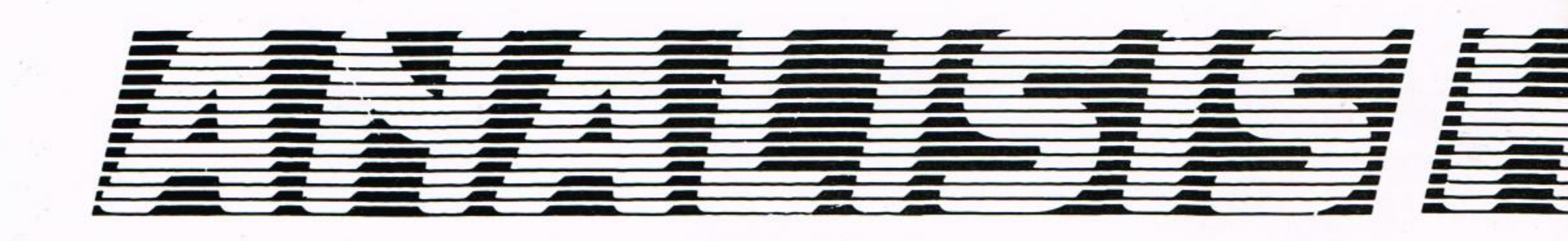
FOTOMECANICA
IMAGEN

IMPRIME
GRAFICAS IBARRA

DISTRIBUCION
DISPRENSA
Eduardo Torroja, 9

Depósito legal: M-10.328-1985

Se solicitará control O.J.D.



PACESEATTER

«EL LENTORRO»

Este mes nos ocupa un extraño artilugio al que podemos llamar «El Lentorro», alias «PACESETTER», aparato que nos va a permitir ralentizar cualquier tipo de juego que caiga en las entrañas de nuestro «SPECTRUM», apropiado para cualquier «lento» de reflejos, para «quedar bien» con nuestros amiguetes presumiendo de fantásticas puntuaciones o bien para ganar alguno de los concursos que pululan por el mundillo del «SPECTRUM», consistentes en llegar al final de un juego o conseguir la puntuación más alta posible.

¿Qué cómo se hace? Muy fácil, solo tenéis que conectar el «interface» como siempre (despacio y con el «SPECTRUM» desenchufado) manipularle en la forma que os indicaremos más adelante, y elegir la velocidad que más se acomode a vuestros reflejos.

Palabra que no os engaño, con este artilugio se puede jugar teniendo los gráficos prácticamente parados (al principio, creí que estaban totalmente parados) hasta la velocidad normal, sólo hay que accionar el potenciómetro que posee «El Lentorro» en su costado derecho y a jugar...

Añadido a esto, el aparato en cuestión es un «interface» programable para «joystick» (con alguna que otra facilidad que os contaré después) de padres británicos, concretamente, está fabricado por «NIDD VALLEY MICRO PRODUCTS LTD.», y se llama como os dije antes «PACESETTER».

Sin más rodeos, os diré que el aparato viene con unas completas, aunque poco abultadas instrucciones, pero que desgraciadamente están en inglés, así que os contaré más o menos los pasos a seguir para hacerlo funcionar.

Como siempre (nunca me cansaré de recordároslo) debemos desenchufar el «SPECTRUM» para conectarle el «interface»; a continuación conectaremos el «joystick» a través de un conector estandard de nueve puntas; en este momento, si no vamos a conectar más periféricos a nuestro socio, a través de la ranura de expansión trasera del «PACESETTER», ya estaremos en condiciones de alimentar a nuestro amiguete con unos cuantos voltios.

Probaremos si el «PACESETTER» está en condiciones, pulsando el botoncito rojo (que hace el papel de interruptor o activador de la parte del «interface» que regula la velocidad de ejecución) sabiendo que está en «ON» si se enciende el «LED» de color rojo colocado encima del «interface».

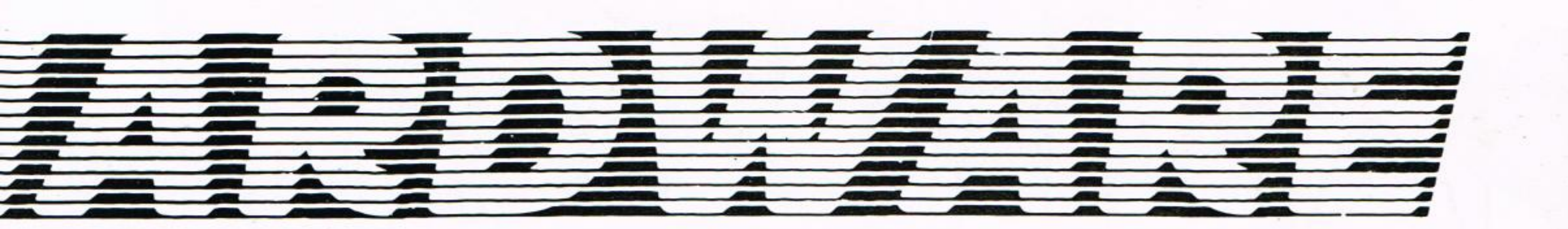
En varias ocasiones, el fabricante nos avisa que el «interface» debe estar apagado para las operaciones de «carga» y «descarga». Así pues, una vez apagado el «PACESETTER», podemos cargar el programa que acompaña al aparato, o si se quiere hacer la prueba que nos aconseja el fabricante, por la que, mediante un corto programa «BASIC» de tres instrucciones, que teclearemos nosotros, se rellenará de números la pantalla, pudiendo controlar la velocidad de impresión mediante el ya mencionado potenciómetro. Al efectuar esta prueba fue cuando pareció que se detenía la ejecución del programa por completo, pero mientras leía las instrucciones, la pantalla fue cambiando de forma casi imperceptiblemente, en la que casi se podía apreciar cómo se escribían los números «PIXEL» a «PIXEL».

Volviendo al programa que acompaña el «interface», si cargamos el programa estaremos en disposición de poder programar el «interface». Para esto, una vez termine la operación de carga, se nos presentará un menú de seis opciones que os comentaré brevemente a continuación.

1) Lista de juegos en librería

Nos imprime en pantalla o por impresora, los juegos, o mejor dicho, los nombres de los juegos con que hemos almacenado en la librería, las distintas combinaciones de teclas, con las que funcionará el «joystick».

El programa o, más propiamente dicho, la librería, contiene algunas combinaciones ya grabadas, como son las del cursor, como el AGF y PROTEK; las dos combinaciones standard, usadas por Sinclair (1 2 3 4 5 y 6 7 8 9 Ø), así como algún juego como «MANIC MINER», «JET PAC», etc.



2) Dar de alta un juego en la librería

Permite añadir nuevos juegos y combinaciones de teclas a la librería de una forma sencillísima; primero se nos preguntará el nombre del juego (admitiendo hasta diez caracteres) y a continuación cada una de las direcciones del juego; efectuando a continuación un «test» de las direcciones almacenadas. Por último, dos detalles sobre la programación; si alguna de las direcciones del «joystick» no se usan, se tecleará «N», y además, el programa está preparado para aceptar dos botones de disparo.

3) Dar de baja un juego en la librería

Sirve evidentemente para dar de baja un juego que ya no nos interesa mantener en la librería.

4) Grabación de la librería

En la librería «caben» de veinte a doscientas cincuenta combinaciones, según se disponga de un «SPECTRUM» de 16 ó 48 K, por ello, se nos ofrece la posibilidad de salvar la librería (cada vez que hayamos añadido uno o varios juegos) en cinta o en «microdrive».

5) Usar un juego de la librería

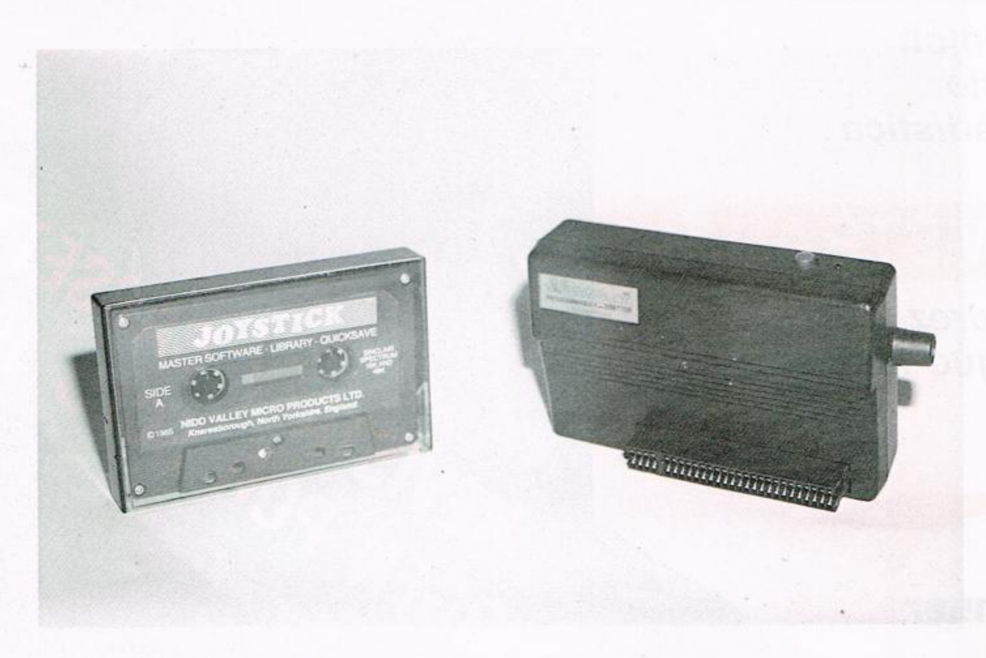
Para esto, simplemente hay que teclear el nombre del juego, siendo programado automáticamente el «interface», y borrando a continuación el programa y la librería de la memoria, para dar cabida al juego, pero sin perder la programación del «interface».

6) Carga rápida

Existe la posibilidad de «salvar» en modo «turbo» cualquier combinación de teclas de un juego, y la rutina de programación del «interface», de una forma un tanto laboriosa, que creo sinceramente no merece la pena detallar en esta breve descripción del periférico, únicamente os diré que permite cargar la rutina de programación de sus teclas, en tan sólo quince segundos, pero de una forma individualizada. Por tanto, a no ser que se grabe una combinación por cinta (saldría carísimo), si tenemos que «rastrear» una cinta con varios juegos, se tardará quizá más que cargando la librería estandard. Por supuesto me refiero a los que no poseáis «microdrive» pues estos últimos no tendrán ningún problema.

No obstante, la verdadera utilidad de esta carga rápida, está en grabar la rutina y su combinación de teclas, antes del juego en las «copias de seguridad» que podáis hacer de los mismos, así, cada vez que vayamos a cargar un juego, por muy pocos segundos más, habremos cargado a la vez la programación del «interface».

En resumen, un curioso invento que a más de uno le va a permitir apreciar esos juegos «aburridísimos» en que era imposible pasar de la primera pantalla porque siempre le «mataban», a otros les permitirá clasificarse en alguna prueba de atletismo, y a uno, de momento le ha servido para conseguir «ver» la pelota en «finales» jugando el «MATCH POINT», e incluso ponérselo difícil a Mac'Enroe.



TYPE AND RUN

LA PRIMERA RE... ...SSETTE PARA TECLEAR Y GRABAR

Pídala en su kiosko o en tiendas especializadas

SUSCRIPCIONES: MONSER, S.A. C/ Argos, 9 - 28037 MADRID

AÑO I – N.º 5

MENSUAL

195 pts.

LISTADO...

...RUM

...ales
...quash
Mandala
Cosmos

MSX

Crack
Caseta de tiro
Dragster

COMMODORE

Simmon
Oeste
Estadística

AMSTRAD

Ajedrez
Buque fantasma

M. NOVOA

REVISTA Y CASSETTE VIRGEN POR SOLO 195 PTS.



CODIGO MAQUINA

CAPITULO VII
(Continuación)

11.—Experimento (Continuación)

Por último, nos queda por considerar la rutina encargada de exhibir el mensaje, que llamaremos SAMEN, y que consiste en la ejecución de dos funciones: borrado de la pantalla y la exhibición propiamente dicha del mensaje, en el que sucesivamente se van incorporando la letra correspondiente a la nueva tecla pulsada.

Tanto la función de borrar la pantalla, como la de exhibir el mensaje, las llevaremos a cabo mediante las rutinas incluidas en la memoria ROM. Ya hemos visto cual era la rutina de exhibir un mensaje, así como la de abrir un canal, por lo cual únicamente nos resta hablar de la que proporciona el borrado de la pantalla, o sea, la función CLS. Esta rutina se encuentra en la posición 3435 (# 0D6B) y no necesita ninguna condición especial para efectuar su trabajo.

El organigrama de esta rutina quedará como se indica en la figura 1, y la rutina en Assembler.

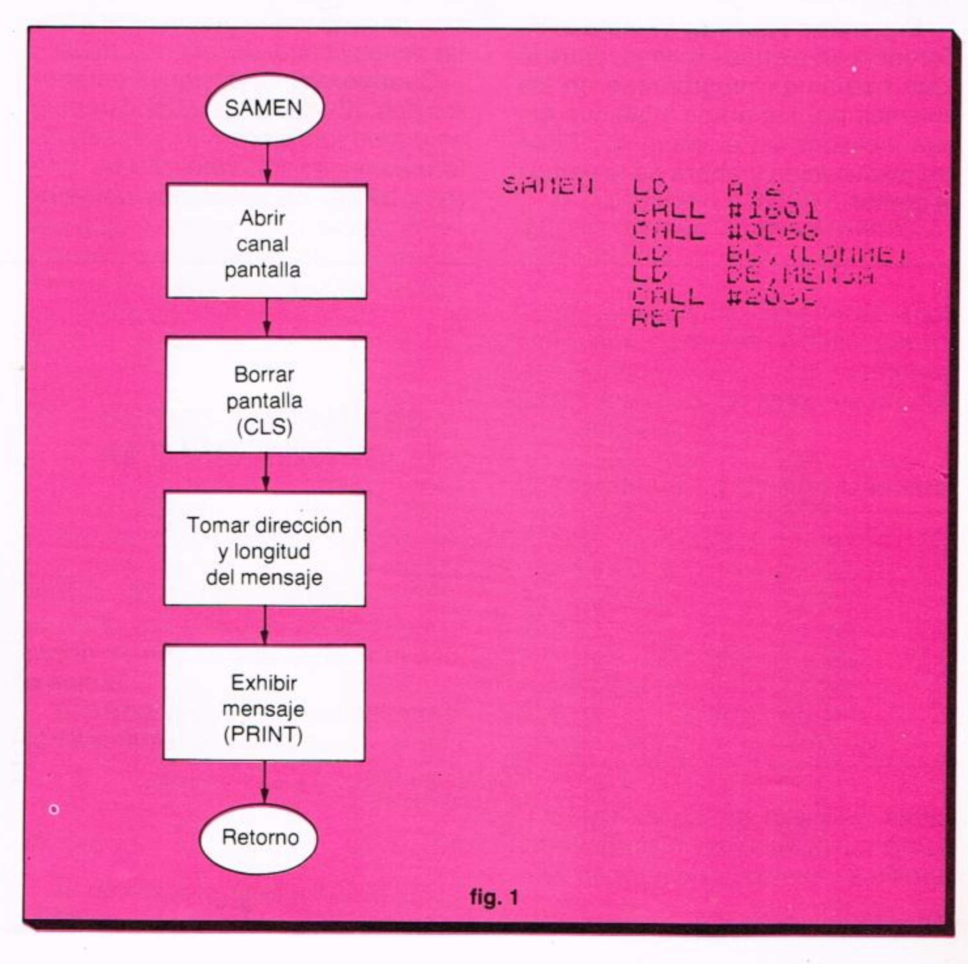
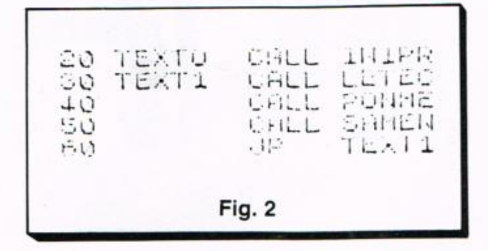


fig. 1

Una vez terminado el análisis y la programación de cada una de las rutinas que conforman cada uno de los bloques del programa no queda más que llevar a cabo el nexo de unión de todas estas funciones a través de las respectivas llamadas CALL. Este programa no considero necesario plasmarlo en un organigrama por lo que únicamente daremos su codificación en Assembler.



```
20 TEXTO   CALL INIPR
30 TEXT1   CALL LETEC
40         CALL PONME
50         CALL SAMEN
60         JP   TEXT1
```

Fig. 2

No considero tampoco necesario dar mayores explicaciones sobre el funcionamiento del mismo. Es demasiado sencillo ¿no?

Ya tenemos todos los eslabones que componen la cadena, o programa. Sólo nos queda introducir todas estas instrucciones en el ordenador a través de un programa Ensamblador. El resultado es algo parecido a lo que refleja la figura 3.

Esperamos que hayáis notado diferencia en algunas rutinas, entre lo que habíamos programado en las mismas por separado y las que están incluidas en el conjunto. Tiene su justificación y ahora os la vamos a contar.

Las diferencias observadas están en función de la rutina TECLA. En efecto, dado que si no se pulsa una tecla no vamos a hacer otra cosa que volvernos a dicha rutina otra vez, nos hemos permitido la osadía de eliminar las dos instrucciones que componían la rutina TECLN, a través de la cual se ponía a cero el Acumulador y se retornaba a la rutina que llamó a ésta. En su lugar, hemos puesto la bifurcación de «no pulsada tecla» (TECLN) tras la apertura del canal.

Como consecuencia de la imposibilidad de que, cuando no se pulsa una tecla, salga de la rutina TECLA, ya no es necesario controlar esta circunstancia en la rutina LETEC, en donde hemos eliminado la instrucción JP NC, LETEC.

Ahora vamos a daros la configuración hexadecimal de las instrucciones para que podáis meterlo a través del programa CARGAHEX a partir de la dirección 4ØØØØ (# 9C4Ø), como podéis observar en la figura 4.

Las cuatro últimas instrucciones del programa parecen un pegote, sin función determinada. En efecto, son un pegote pero tienen una misión muy concreta que vamos a tratar de explicar de forma resumida.

Cuando en instrucciones anteriores nos referíamos a unos campos, en donde teníamos almacenada información, en concreto MENSA, LONME y DIRLE, no habíamos contemplado la forma de crear, o indicar donde se encontraban, esos espacios. Si dichos nombres no hubieran sido encontrados por el programa Ensamblador nos hubiera notificado tal situación anómala, y el programa no había podido ser ensamblado correctamente, y por lo tanto ejecutarse con posterioridad. Hubiéramos obtenido una solución correcta con la sustitución de dichos nombres por direcciones de memoria. Es decir, donde pone MENSA haber puesto 50004, en lugar de LONME poníamos 50002 y, por último, en vez de DIRLE sustituirlo por 50000. Esto es correcto como ya decimos, pero tiene dos inconvenientes: primero, tenemos que elegir las direcciones lo suficientemente alejadas de lo que va a ser el programa que, aunque movamos hacia atrás las instrucciones, no le afecten dichos cambios, pues de lo contrario, al modificar el contenido de los bytes situados en estas direcciones nos estaríamos «machacando» nuestro programa. En segundo lugar, no sería didáctico porque, a mí particularmente, no me dice nada 50000 y por el contrario si sé lo que quiere decir con DIRección de LEtra (DIRLE).

La solución pues, viene de la mano del propio ensamblador. Si damos un nombre simbólico a una instrucción que ocupe el mismo número de bytes que los que yo necesito, es lo mismo que si yo me hubiera creado un campo que lo llamo por ese nombre. Y ese es el objeto de

```
20  TEXTO   CALL  INIPR
30  TEXT1   CALL  LETEC
40          CALL  PONME
50          CALL  SAMEN
60          JP    TEXT1
70  INIPR   LD    HL,MENSA+3
80          LD    (DIRLE),HL
90          LD    HL,3
100         LD    (LONME),HL
110         RET
120 LETEC   CALL  TECLA
130         SUB   13
140         JP    Z,SABAS
150         ADD   A,13
160         RET
170 SABAS   INC   SP
180         INC   SP
190         RET
200 PONME   LD    HL,(DIRLE)
210         LD    (HL),A
220         INC   HL
230         LD    (DIRLE),HL
240         LD    HL,(LONME)
250         INC   HL
260         LD    (LONME),HL
270         RET
280 TECLA   LD    A,1
290         CALL  #1601
300 TECLN   CALL  #028E
310         LD    C,0
320         JP    NZ,TECLN
330         CALL  #031E
340         JP    #NC,TECLN
350         DEC   D
360         LD    E,A
370         CALL  #0333
380         RET
390 SAMEN   LD    A,2
400         CALL  #1601
410         CALL  #0068
420         LD    BC,(LONME)
430         LD    DE,MENSA
440         CALL  #203C
450         RET
460 DIRLE   LD    A,0
470 LONME   LD    A,0
480 MENSA   LD    D,0
490         INC   B
```

Fig. 3

```
Addr Hex        Op    Operand/Notes

9C40 CD4F9C     CALL  9C4F
9C43 CD5C9C     CALL  9C5C
9C46 CD6A9C     CALL  9C6A
9C49 CD939C     CALL  9C93
9C4C C3439C     JP    9C43
9C4F 21AD9C     LD    HL,9CAD
9C52 22A69C     LD    (9CA6),HL
9C55 210300     LD    HL,0003
9C58 22A89C     LD    (9CA8),HL
9C5B C9         RET
9C5C CD7A9C     CALL  9C7A
9C5F D60D       SUB   0D
9C61 CA679C     JP    Z,9C67
9C64 C60D       ADD   A,0D
9C66 C9         RET
9C67 33         INC   SP
9C68 33         INC   SP
9C69 C9         RET
9C6A 2AA69C     LD    HL,(9CA6)
9C6D 77         LD    (HL),A
9C6E 23         INC   HL
9C6F 22A69C     LD    (9CA6),HL
9C72 2AA89C     LD    HL,(9CA8)
9C75 23         INC   HL
9C76 22A89C     LD    (9CA8),HL
9C79 C9         RET
9C7A 3E01       LD    A,01
9C7C CD0116     CALL  1601
9C7F CD8E02     CALL  028E
9C82 0E00       LD    C,00
9C84 C27F9C     JP    NZ,9C7F
9C87 CD1E03     CALL  031E
9C8A D27F9C     JP    NC,9C7F
9C8D 15         DEC   D
9C8E 5F         LD    E,A
9C8F CD3303     CALL  0333
9C92 C9         RET
9C93 3E02       LD    A,02
9C95 CD0116     CALL  1601
9C98 CD6B0D     CALL  0D6B
9C9B ED4BA89C   LD    BC,(9CA8)
9C9F 11AA9C     LD    DE,9CAA
9CA2 CD3C20     CALL  203C
9CA5 C9         RET
9CA6 3E00       LD    A,00
9CA8 3E00       LD    A,00
9CAA 1600       LD    D,00
9CAC 04         INC   B
```

Fig. 4

esas tres instrucciones. Los dos primeros campos necesitan dos bytes por lo que cualquier instrucción que ocupe dos bytes nos serviría para nuestro propósito. Para la tercera, y dado que es la que va a representar el comienzo del campo de mensaje, en el cual vamos a poner en primer lugar el código #16 para que se realice la función AT, como ya comentábamos en ejercicios anteriores. Para ello hemos buscado la instrucción cuyo código de operación correspondiera a dicho valor hexadecimal y hemos comprobado que se trata de la LD D, n que precisamente hemos visto con anterioridad (n.º 3, pág. 8).

CAPITULO VIII

1.—INTRODUCCION

En el ejercicio del capítulo anterior se deslizaron dos instrucciones en las que aún no nos habíamos detenido para explicarlas. Se trata de las instrucciones ADD y SUB, que pertenecían a la rutina LETEC. Aunque al explicar la rutina ya indicábamos la función que realizaban, en este capítulo vamos a detenernos concretamente en ellas. Si este lapsus os ha podido perjudicar la comprensión perfecta de la susodicha rutina, os ruego que perdonéis al autor. No hacía más que tratar de proporcionar una rutina a su juicio interesante, y sólo tras un repaso más concienzudo de lo que había escrito se dio cuenta de tal error. No obstante, en este capítulo espera poder subsanar la falta cometida, y solucionar las lagunas que hayan podido surgir.

También en este capítulo veremos algunas instrucciones de bifurcación, similares a las vistas en el capítulo anterior, para poder abordar en un próximo capítulo la forma de hacer preguntas, y cómo recibir contestación.

2.—Sumas aritméticas de 8 bits (ADD)

Este conjunto de instrucciones lo podemos dividir en dos subconjuntos. Sus funciones son idénticas: Suman una cantidad al registro Acumulador. La diferencia estriba en el modo de decirle cual es la cantidad a sumar.

El primer grupo tiene como formato Assembler el siguiente:

ADD A,p

en donde,

A representa al registro Acumulador, y

p representa a alguno de los registros simples, o una constante 0 y 255 decimal.

En la siguiente tabla mostramos las instrucciones Assembler que corresponden a este subconjunto así como los Códigos Máquina en que se traducen.

Assembler	Código máquina
ADD A,A	87
ADD A,B	80
ADD A,C	81
ADD A,D	82
ADD A,E	83
ADD A,H	84
ADD A,L	85
ADD A,n	C 6.xx

Como se puede apreciar, todas las instrucciones ocupan un único byte de memoria, excepto la última,

(Continúa en pág. 12)

JUEGO DEL MES

PISTOLERO

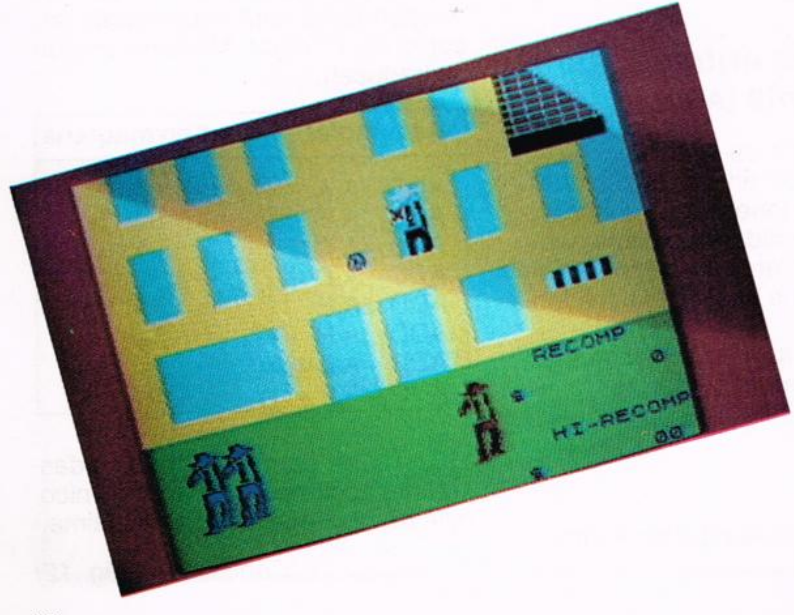


Con unas viejas botas, de indefinido color gracias al polvo, apoyadas sobre la mesa, leía el viejo SHERIFF los papeles de WANTED que le acababan de llegar en la última diligencia.

La peligrosa banda de James, reclamada en todo el territorio nacional, había tomado dirección Sur después del último atraco.

Uno a uno, fue leyendo detenidamente cada papel con la fotografía impresa de todos los miembros de la banda James y su amplio historial; más de ochenta muertos quedaron tendidos detrás de sus setenta y siete asaltos.

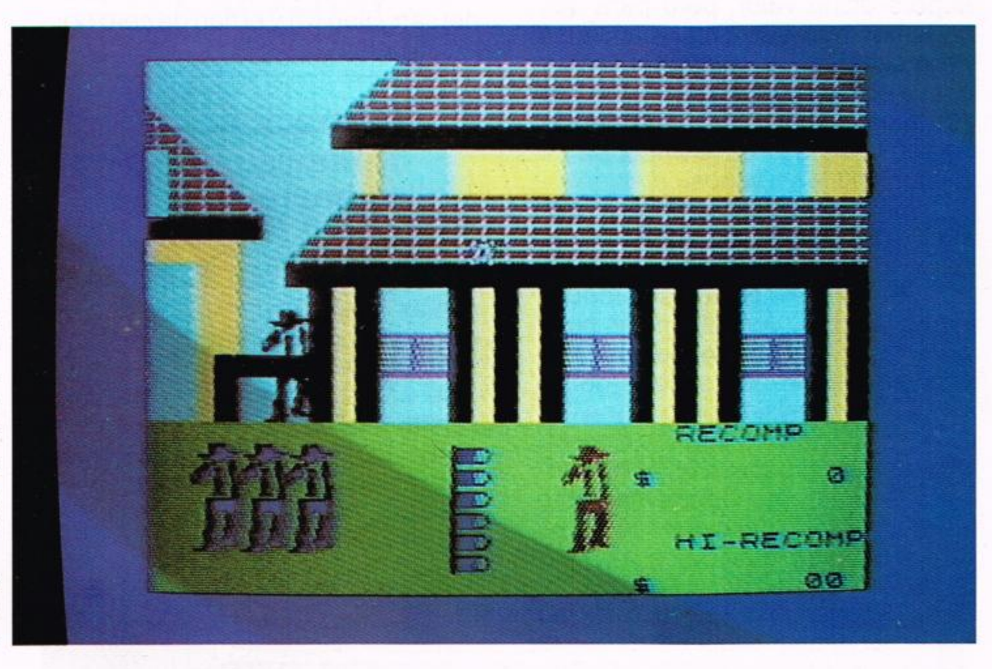
La memoria del viejo Sheriff Thomas voló con dificultad en el tiempo, como cinco años atrás, cuando puso a la sombra a James y a otros

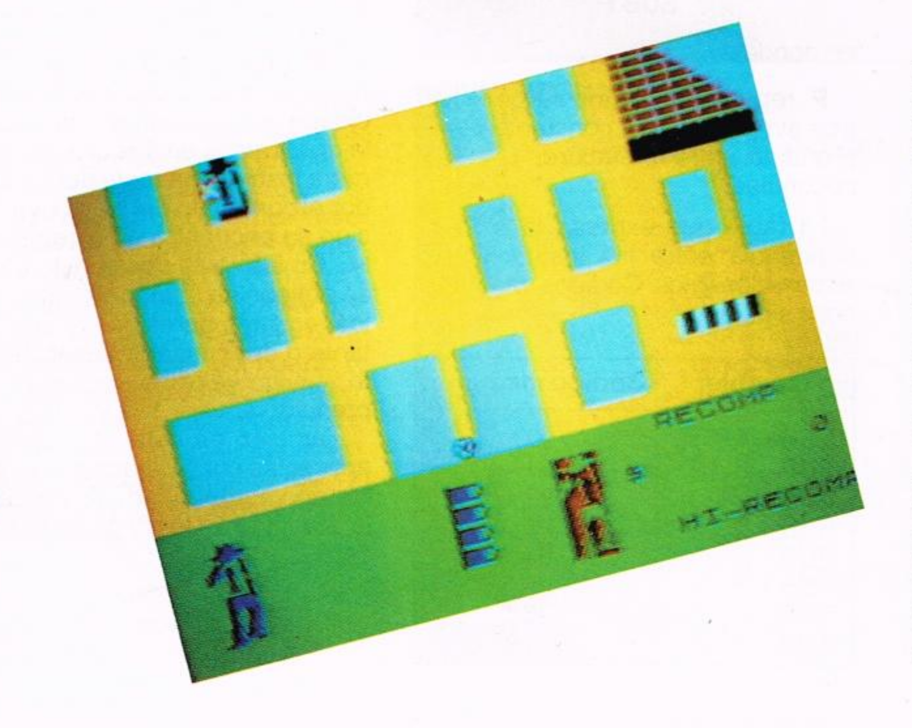




dos miembros de su sanguinario grupo. Aquel oscuro día los James le juraron volver para matarle y parecía que estaban dispuestos a cumplir su promesa. Pero el viejo Thomas no era persona que se amilanase con facilidad y además ciertamente él ya había superado con mucho el límite de edad que aparecía grabado sobre una sucia losa de piedra en el cementerio de cualquier pueblo..., en esas losas solía acompañar el epitafio: «El pueblo de... al valiente sheriff... que supo dar su vida a la edad de... en defensa de la ley y el orden», y el viejo Thomas nunca había leído que en ninguno de éstos la edad superase los 40.

Su intuición era aún lo bastante buena como para saber que cuando los James llegasen al pueblo no debería de quedar nadie en las calles, aunque seguro que Tom el del Saloon y Moore el del Almacén querrían acompañarle en la lucha, pero esto era algo casi personal; era cuestión de amor propio, y además ¡qué caramba! ya iba siendo hora de que un experimentado tirador como tú les dé una lección a esos estúpidos y desalmados pistoleros y saliendo al centro de la calle, te dispones a acabar con ellos de una vez por todas.





Recuerda que para girar (cambiar tu ángulo de visión) deberás pulsar la tecla CAPS SHIFT, para subir el punto de mira utiliza la tecla 1, para bajarlo la A, para la izquierda la Q, para la derecha la P y para disparar, utiliza la última fila, de Z hasta BREAK.

(Viene de página 9)

que ocupa dos, pues además de llevar el código de operación también incluye la configuración hexadecimal de la constante a sumar.

El funcionamiento de estas instrucciones es sencillo, pues consiste en realizar la suma del contenido del Acumulador con la constante, o el contenido del registro simple especificado en la instrucción, y el almacenamiento del resultado en el registro Acumulador.

Cuando el resultado de estas sumas supera el valor de 255, máximo que puede representarse en un solo byte, entonces en el Acumulador sólo se almacena la cantidad que supera dicho valor, pero para que sepamos que esa circunstancia se ha presentado, el microprocesador activa el flag de acarreo, o carry flag (C), en el registro F, el cual podemos analizar con las instrucciones de bifurcación que ya vimos.

Cuando el resultado de una operación aritmética sobrepasa la capacidad del campo donde debe almacenarse, se dice que se produce un «overflow», vocablo inglés cuya traducción sería «inundación» y que expresa lo que sucede de un modo algo más simbólico.

Del mismo modo, cuando el resultado que se almacena en el Acumulador es cero, independientemente de la activación del carry-flag (C) se activa también el bit de cero (Z) del registro F.

En el segundo grupo veremos las instrucciones de suma en las que el byte a sumar al Acumulador no está referenciado tan directamente como en las anteriores. Estas instrucciones tienen este formato general.

ADD A, (rd)

en donde:

A es el registro Acumulador, como siempre, y

rd representa al registro par HL, y a los registros Ix e Iy a los cuales se les suma además una constante denominada «desplazamiento».

La tabla de instrucciones Assembler y los correspondientes códigos máquina son:

Assembler	Código máquina
ADD A,HL	86
ADD A (Ix + d)	DD 86 xx
ADD A,(Iy + d)	FD 86 xx

La explicación del funcionamiento de estas instrucciones vamos a dividirlo también en dos partes, para comentar los matices propios de los dos subtipos.

La instrucción ADD A, (HL) hace que el byte señalado por la dirección contenida en el registro par HL sea el que se sume al Acumulador.

En las otras dos instrucciones la dirección del byte a sumar al Acumulador se obtiene sumando el contenido del registro para Ix, ó Iy, el valor constante d que es un número comprendido entre −128 y +127.

También en este grupo de instrucciones debemos aplicar las consideraciones expuestas anteriormente con respecto a la activación del carry flag (C) y el flag de cero (Z).

3.—Sustraciones aritméticas al acumulador (SUB)

Del mismo modo que cuando hablamos en el párrafo anterior de las sumas, este conjunto de instrucciones podemos dividirlo en otros dos subconjuntos, que también cumplen idéntica función: Restar una cantidad al contenido del registro Acumulador. Como en el caso anterior, la diferencia existente entre los dos subconjuntos consiste en la forma de indicarle al microprocesador cuál es la cantidad a restar.

El primer subconjunto tiene como formato Assembler el siguiente:

SUB P

en donde:

P representa alguno de los registros simples, o una constante comprendida entre los valores 0 y 255 decimales.

La tabla que expresa la correspondencia entre las instrucciones Assembler y los Códigos Máquina es la siguiente:

Assembler	Código máquina
SUB A	97
SUB B	91
SUB C	92
SUB D	93
SUB E	94
SUB H	95
SUB L	96
SUB n	D6 xx

Todas las instrucciones son de un solo byte de longitud, excepto la última que es de 2 bytes.

La cantidad a restar del contenido del Acumulador viene dado por la constante, o por el contenido del registro simple que figura en la instrucción. Cuando el número a restar es superior al contenido del Acumulador se activa el flag (C) o carry-flag, indicando que ha habido desbordamiento. Si el resultado de la sustración es cero, el bit del registro F que se activa es el que corresponde al cero (Z).

El segundo grupo de instrucciones está compuesto por aquellas en las que el byte a sumar esté referenciado de un modo indirecto. Estas instrucciones tienen el siguiente formato:

SUB (rd)

en donde,

rd representa al registro par HL, y a los registros pares Ix e Iy, a los cuales se les suma una constante denominada «desplazamiento».

La tabla de las instrucciones Assembler que componen este subgrupo, así como sus correspondientes Códigos Máquina es la siguiente:

Assembler	Código máquina
SUB (HL)	96
SUB B(Ix + d)	DD 96 xx
SUB (Iy + d)	FD 96 xx

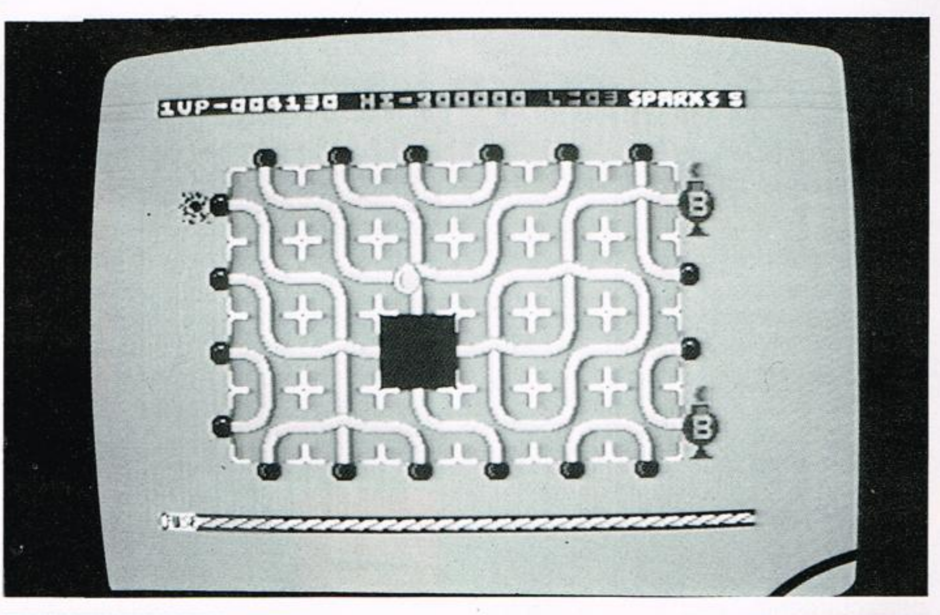
Del mismo modo que en el caso de las sumas, la explicación del funcionamiento de estas instrucciones la dividiremos en dos grupos. La primera instrucción resta del Acumulador el contenido del byte cuya dirección se encuentra en el registro par HL. En las otras dos el byte a restar se obtiene de sumar al contenido de los registros pares Ix ó Iy una constante d denominada desplazamiento, y cuyo valor se encuentra comprendido entre −128 y +127.

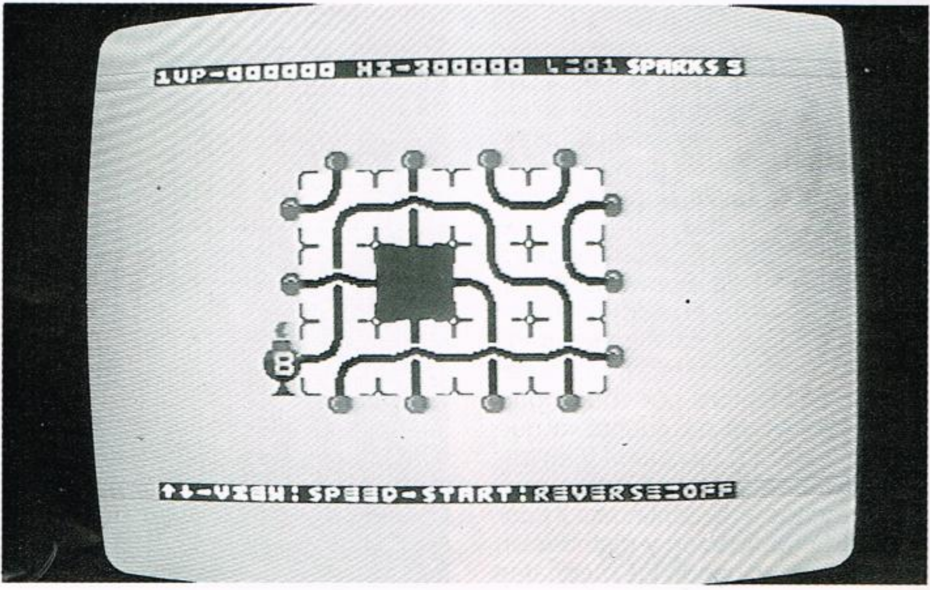
También en este caso debemos aplicar las consideraciones expuestas con anterioridad con respecto a la actividad de los flags de acarreo y cero (C y Z).

CONFUZION

Cuando parecía que todo estaba inventado en lo que respecta a los juegos no arcade (WAR GAMES, MESA-SALON, AVENTURAS...) llega hasta nosotros el programa CONFUZION, de la firma INCENTIVE. Ya conocíamos de su existencia por la publicidad que en las revistas especializadas del Reino Unido veníamos viendo desde hace unos meses, en ellas rezaba la frase «LA FUSIÓN DE LA MENTE Y LA MÁQUINA», pero esto no era suficiente para imaginarse en que consistía el juego. Pues bien, nos encontramos ante un juego vivo, excitante, que lo cierto es que anda a caballo entre el arcade y el salón, tomando de estos dos tipos de juegos, los mejores condimentos para elaborar un juego que te dejará un buen sabor de boca, aunque en ningún momento debes esperar un juego fácil, deberás emplearte a fondo, concentrar tu mente antes de empezar, tensar tus músculos y afinar tus reflejos para en una sentada pasar del cuarto o quinto nivel, nuestro especialista en juegos (al que no le faltan horas de vuelo) no consiguió pasar del sexto en una sentada de cuatro horas.

La cosa consiste en algo tan «fácil» como hacer que la chispa que corre por la mecha, alcance la bomba que aparece en el lado opuesto, pero os aseguro que de fácil no tiene ni un pelo.





ANALISIS SOFTWARE

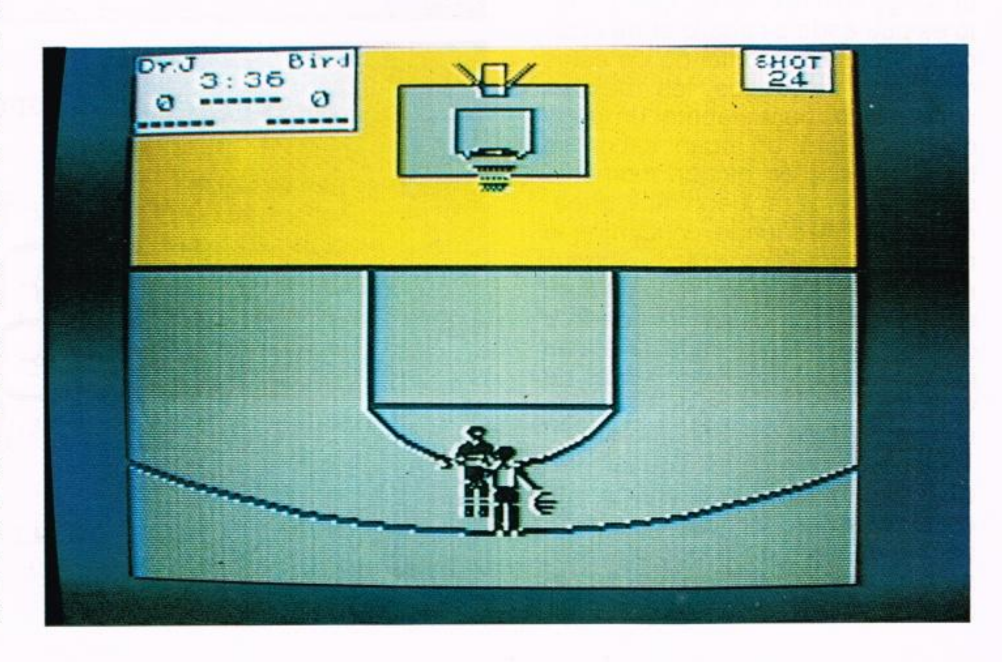
ONE ON ONE

Ya echábamos en falta, con la euforia de los programas de simulación deportiva que invaden el mercado, que no hubiera alguien que se atreviera con un partido de baloncesto, la empresa era difícil, lo reconocemos, pero no imposible.

Hace algún tiempo (no demasiado) que los programas que se hacían expresamente para nuestro querido ordenador, eran convertidos, después de comprobar su éxito, a otros sistemas, como Commodore 64, MSX, AMSTRAD, etc. y lógicamente, en el momento de la conversión (sobre todo si los fines son meramente lucrativos), se perdía en calidad. Todos los programas, se hacen en y para un ordenador específico, por alguien que lo conoce perfectamente, y que por lo tanto sabe aprovechar totalmente las características de este aparato, llegando incluso, movido por el cariño hacia su idea, a sacar partido de los propios defectos de su maquinita, siempre como ya he dicho, movido por el cariño hacia una idea. Una vez terminado este programa, se ofrece a las casas que se dedican a la distribución de Software, éstas, tras unos vistazos al mismo deciden si se lo quedan o no. Supongamos que sí. La firma compra los derechos (casi siempre por cuatro gordas, si tenemos en cuenta lo que cuesta hacer un programa bien hecho), decíamos que la firma compra los derechos y a continuación comienza la fabricación y distribución del producto. Cuando los magnates de la industria del Software han comprobado total y absolutamente que el programa es un éxito, no quieren quedarse con la tarta incompleta aunque dispongan ya de la mayor parte, y por un módico precio, el juego se convierte a otro sistema, pero ya sin el cariño de aquel programador que hizo el primero, y además teniendo que conservar la estructura del mismo.

De esta montaña de impersonalidades surge el nuevo juego en otro sistema... y digo «nuevo juego» en el más puro y literal sentido de la expresión, pues el resultante no tiene apenas nada que ver con el original..., pero al fin y al cabo, la industria no tiene nada que ver con los sentimientos.

Pues bien, amigos, esto es lo que ha pasado con este programa, nosotros lo hemos visto en el sistema original y después lo recibimos convertido a Spectrum..., qué queréis que os digamos, la casa ARIOLA SOFT da la impresión de haber hecho realidad todo lo anteriormente expuesto; aunque quien no haya visto el programa original pueda parecerle no malo del todo, si los comparas, !puff!



Dibujador

Este interesante programa es una herramienta muy útil para dibujar en el Spectrum. Con él podrás hacer diseños y dibujos bastante completos.

Instrucciones

Primero se te preguntará por brillo, intermitencia y transparencia. En cada uno de los casos deberás responder con 0 (No) o 1 (Sí).
Después se te preguntará por tinta, papel y borde, respondiendo en cada uno de los casos con un número del 0 al 17.
En las opciones «C» y «P», el programa responde a las teclas 5, 6, 7 y 8.
La «F» vuelve al menú principal.

Opciones

D: Define nuevo carácter (fila a fila y en binario).
R: Repite un caracter definido anteriormente.
RT: Repite un caracter definido anteriormente variando su tinta, papel, etc.
C: Escribe caracter a caracter.
P: Escribe pixel a pixel.
SAVE: Almacena la pantalla que estamos viendo.
LOAD: Carga una pantalla para trabajar en ella.
TEXTO: Permite escribir texto donde y como quieras.
CLS: Borra la pantalla.

NOTA: Los comandos SAVE, LOAD, TEXTO y CLS se introducen por medio del teclado y con todas las letras.
A veces puede ocurrir que se salga del programa. Si desea volver a él sin destruir la pantalla que estaba haciendo, de debe oprimir: GO TO 140.

```
   10 REM ***********************
   11 REM *  programa DIBUJADOR *
   12 REM *   © Mario Alvarez   *
   13 REM *         y           *
   14 REM *      MONSER S.A.    *
   20 REM ***********************
   25 POKE 23609,30
   30 INPUT "color del borde ?? "
;borde: BORDER borde
   40 INPUT "color del fondo ?? "
;fondo: PAPER fondo: CLS
   50 INPUT "codigo opcion ?? ";o
$
   60 IF o$="load" THEN GO TO 600
   61 IF o$="save" THEN GO TO 500
   62 IF o$="c" THEN GO TO 300
   63 IF o$="p" THEN GO TO 400
   64 IF o$="d" THEN GO TO 100
   65 IF o$="r" THEN GO TO 200
   66 IF o$="rt" THEN GO TO 250
   68 IF o$="cls" THEN CLS
   70 IF o$="texto" THEN GO TO 70
0
   90 GO TO 50
   96 REM
   99 REM ***********************
  100 REM **definO nuevo caracter
  101 REM ***********************
  102 REM
  110 FOR n=0 TO 7
  120 INPUT "fila ";(n);" :";f$
  121 LET fila=(VAL f$(1 TO 1)*12
8)+(VAL f$(2 TO 2)*64)+(VAL f$(3
 TO 3)*32)+(VAL f$(4 TO 4)*16)+(
VAL f$(5 TO 5)*8)+(VAL f$(6 TO 6
)*4)+(VAL f$(7 TO 7)*2)+VAL f$(8
 TO 8)
  122 POKE USR "p"+n,fila
  125 NEXT n
  144 REM
  145 REM ***********************
  146 REM   defino variables impr
  147 REM ***********************
  148 REM
  150 INPUT "tinta ??";t
  152 INPUT "papel ??";p
  154 INPUT "brillo ??";b
  156 INPUT "intermitente ??";f
  158 INPUT "transparente ??";o
  160 INPUT "coord. horiz.  ";h
  162 INPUT "coord. vert.  ";v
  165 PRINT AT h,v; INK t; PAPER
p; BRIGHT b; FLASH f; OVER o;CHR
$ 159
  190 GO TO 50
  200 REM
  201 REM ***********************
  202 REM rep. char. otras coord.
  203 REM ***********************
  204 REM
  205 GO TO 160
  250 REM
  251 REM ***********************
  252 REM rep. char. otras variab
  253 REM ***********************
  254 REM
  255 GO TO 150
  300 REM
  301 REM ***********************
  302 REM imprimo char. a char.
  303 REM ***********************
  304 REM
  305 FOR n=0 TO 7: POKE USR "q"+
n,0: NEXT n
  310 INPUT "coor. horiz. ?? ";h
  315 INPUT "coor. vert. ?? ";v
  320 INPUT "tinta ?? ";t: INPUT
"papel ?? ";p: INPUT "brillo ??
";b: INPUT "intermitente ?? ";f:
 INPUT "transparente ?? ";o
  330 PRINT AT h,v; INK t; PAPER
p; BRIGHT b; FLASH f; OVER o;CHR
$ 160
  350 IF INKEY$="5" THEN LET v=v-
1
  351 IF v<0 THEN LET v=0
  352 IF INKEY$="8" THEN LET v=v+
1
  353 IF v>31 THEN LET v=31
  354 IF INKEY$="6" THEN LET h=h+
1
  355 IF h>21 THEN LET h=21
  356 IF INKEY$="7" THEN LET h=h-
1
  357 IF h<0 THEN LET h=0
  360 IF INKEY$="f" THEN GO TO 50
  370 GO TO 330
  400 REM
  401 REM ***********************
  402 REM dibujo pixel a pixel
  403 REM ***********************
  404 REM
  410 INPUT "horiz. de 0 a 175 :
";h
  411 INPUT "vert. de 0 a 255 : "
;v
  415 INPUT "tinta ?? ";t
  416 INPUT "brillo ?? ";b
  420 PLOT BRIGHT b; INK t;v,h
```

```
430 IF INKEY$="5" THEN LET v=v-
1
431 IF v<0 THEN LET v=0
432 IF INKEY$="8" THEN LET v=v+
1
433 IF v>255 THEN LET v=255
434 IF INKEY$="6" THEN LET h=h-
1
435 IF h<0 THEN LET h=0
436 IF INKEY$="7" THEN LET h=h+
1
437 IF h>175 THEN LET h=175
438 IF INKEY$="f" THEN GO TO 50
440 GO TO 420
500 REM
501 REM **********************
502 REM ** salvo la pantalla **
503 REM **********************
504 REM
505 INPUT "nombre : ";n$
510 SAVE n$SCREEN$
520 GO TO 50
600 REM
```

```
601 REM **********************
602 REM ** cargo la pantalla **
603 REM **********************
604 REM
605 INPUT "nombre : ";n$
610 LOAD n$SCREEN$
620 GO TO 50
700 REM
701 REM **********************
702 REM ** imprimo textos  ****
703 REM **********************
710 INPUT "texto ? ";t$
720 INPUT "tinta ? ";t
722 INPUT "papel ? ";p
724 INPUT "intermitente ? ";f
726 INPUT "transparente ? ";o
728 INPUT "brillo ? ";b
730 INPUT "coord. horiz. ? ";h
735 INPUT "coord. vert. ? ";v
740 PRINT AT h,v; INK t; PAPER
p; FLASH f; BRIGHT b; OVER o;t$
750 GO TO 50
```

Bombero

```
           notas graficas

             A B C D E
            ===========

  1 REM       BOMBERO
  2 REM © Carlos Martin
  3 REM          y
  4 REM    MONSER S.A.
  5 REM
  6 REM    mov. derecha = j
  7 REM    mov. izda.   = s
  8 REM
  9 GO SUB 41
 10 GO SUB 74
 11 GO SUB 44
 12 FOR r=1 TO 10
 13 GO SUB 35
 14 GO SUB 57
 15 GO SUB 38
 16 REM caida bomba
 17 FOR n=0 TO 21
 18 PRINT AT n-1,a;" "
 19 LET a=a+((INKEY$="j")-(INKE
Y$="s"))
 20 IF a>31 THEN LET a=31
 21 IF a<0 THEN LET a=0
 22 PRINT AT n,a;""
 23 BEEP .01,65
 24 NEXT n
 25 IF a<>c+1 THEN GO SUB 69
 26 IF a=c+1 THEN GO SUB 60
 27 PRINT #1;AT 1,3;"PUNTOS = "
;puntos;AT 1,16;"BOMBAS = ";r
 28 PRINT AT 10,8; FLASH 1;" PU
LSE UNA TECLA "
 29 PAUSE 5: PAUSE 0
 30 PRINT AT 10,8; FLASH 0;"
               "
 31 PRINT AT n-1,c;"___"
 32 PRINT AT 21,a;" "
 33 NEXT r
 34 STOP
 35 REM aleatorio destino
 36 LET c=INT (RND*30)
 37 RETURN
 38 REM aleatorio salida bomba
 39 LET a=INT (RND*30)
 40 RETURN
```

```
 41 REM inicializo variables
 42 LET puntos=0
 43 RETURN
 44 REM dibujo el paisaje
 45 PAPER 0
 46 INK 7
 47 CLS
 48 BORDER 0
 49 FOR n=0 TO 31
 50 LET b=INT (RND*21)
 51 PRINT AT b,n;"."
 52 NEXT n
 53 FOR n=0 TO 31
 54 PRINT AT 21,n;""
 55 NEXT n
 56 RETURN
 57 REM colocacion destino
 58 PRINT AT 21,c;""
 59 RETURN
 60 REM rutina de exito
 61 FOR x=55 TO 56
 62 FOR y=10 TO 1 STEP -1
 63 BEEP .01,y
 64 BEEP .01,x
 65 NEXT y
 66 NEXT x
 67 LET puntos=puntos+1
 68 RETURN
 69 REM rutina de fracaso
 70 FOR x=200 TO 255
 71 OUT 254,x
 72 NEXT x
 73 RETURN
 74 REM redefino caracteres
 75 RESTORE
 76 FOR z=0 TO 39
 77 READ byte
 78 POKE USR "a"+z,byte
 79 NEXT z
 80 DATA 90,90,60,126,122,102,6
0,24
 81 DATA 0,0,3,12,51,207,127,12
7
 82 DATA 60,195,60,255,195,195,
195,195
 83 DATA 0,0,192,48,204,243,254
,254
 84 DATA 0,0,16,20,12,136,153,2
55
 85 RETURN
```

TABLON

Se precisan distribuidores en todo el territorio nacional para productos de primeras marcas de:

- Computer
- Video
- Audio

Dirigirse a:

MONSER, S.A.
C/ Argos, 9
28037-Madrid
Telf. 742 72 12/96 (Srta. Yolanda)

Se vende ordenador SINCLAIR SPECTRUM 48K. Nuevo.
Paco Márquez
C/ Las Musas, n.° 13
28022-Madrid
Telf.: 741 57 03

Ocasión:
Vendo cintas para SPECTRUM con 15 programas (BRUCE LEE, DELTA WINGT, PROFANATION, etc.) por 1.500 ptas.
Rubén
Telf.: (983) 23 30 70
VALLADOLID
(Llamar horas de comida.)

Vendo por sólo 5.000 ptas., o cambio por cualquier periférico u órgano electrónico: gran fichero conteniendo curso de introducción a la informática y lenguaje Basic. Además regalo cinta con últimas novedades, catálogos ordenadores, revistas, libros, discos, etc.
Fernando Castán
C/ Padre Manjón, n.° 34, 5.° E
50010-ZARAGOZA
Telf.: (976) 34 83 59

Club de Oviedo, con Sección de Informática (MSX, SPECTRUM, AMSTRAD, ORIC), desea ampliar el número de socios.
Dirigirse a:
Alejandro Fernández Brizuela
Comandante Caballero, 4, 7.° D
33005-OVIEDO

QUEREMOS VER TUS PROGRAMAS BASIC: SPECTRUM, C64, MSX Y AMSTRAD

Sabemos que eres lo suficientemente ingenioso para no necesitar copiarlos de nadie. Mándanos una cinta de cassette con tu programa, y a ser posible, un listado del mismo.

Premiamos con 5.000 ptas. y un ejemplar de la revista en cuestión, cada programa que publiquemos en cualquiera de nuestras cinco publicaciones.

Envíanos el tuyo, cuanto antes lo hagas, más posibilidades tendrás de verlo publicado.

MONSER, S.A.
C/ Argos, 9
28037-MADRID

Deprisa, deprisa...


sinclair
Spectrum
ORDENADOR EDUCATIVO
ORDEDU
Monser
REVISTA MENSUAL
AÑO I - N.º 7
LA MEJOR REVISTA CON CASSETTE
P.V.P.
495
CLOWN
SIMETRIA
INTEGRALES
Programa Basic
LEY DE NEWTON
Pídala en su kiosko o en tiendas especializadas
SUSCRIPCIONES: MONSER, S.A. C/ Argos, 9 - 28037 MADRID
Comentario Soft
ANALISIS

MONSER cada dia +



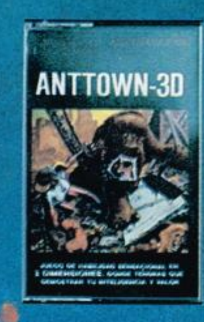
6 SuperCasettes Full Memory por solo 1.795 pts.



Libreto que contiene manual de instrucciones en Castellano +Aplicaciones para tu Spectrum













Ya a la venta en Kioscos, Tiendas Especializadas y Departamento de Informatica del El Corte Inglés



MONSER S. A.
C/ Argos, 9 - 28037 Madrid.
Teléfonos: (91) 742 72 12 - 742 72 96